evaluation system for SAB80C166 (2)

part 2: construction and use

Despite its apparent complexity, the practical circuit of the SAB80C116 evaluation system is only marginally different from typical 8-bit systems as published in Elektor Electronics.

Design by K.-H. Domnick

Just like almost any other microcontroller system, the 80C166 16-bit board is marked by buses for data, address and control signals. Because the databus of the 80C166 has a width of 16 lines, many peripheral ICs like latches and memory elements have to be duplicated. The 80C166 offers a total of six ports (P0-P5), of which P5 is employed by the 8-channel A-D converter. The corresponding lines are protected against voltage surges by zener diode-resistor networks R10-R19, R21/22/D3-D13.

Port P0 is multiplexed as the combined address and data bus. The actual address bus starts behind two latch components, IC9-IC10. The bus has an addressing capacity of 64 kBytes. The address bus is routed to memory devices IC11-IC18, GAL IC4, Real Time Clock IC5 and decoder IC6. Moreover, it is made available on extension connector K2 for use by external circuits.

The digital inputs and outputs are taken to connector K1 by way of buffer devices IC19-IC22. The level conversion to RS232 signals is carried out in IC1 and IC2.

Resistors R1-R4 are optional, and intended for special applications of the SIOs (serial input/output devices). Circuit IC7 is also optional for unusual applications employing the controller's Ready signal and a lengthened output signal on port line P1.13.

Because the chip select signals are decoded in a GAL, it is possible to modify the memory configuration.

Two field-effect transistors (FETs) type BS170 (T1 and T2) are included to ensure that the current through the pull-up resistors disappears when the board is switched off with the RAMs still powered by the backup battery.

After any write operation to the EEPROM, a pause of 2-4 ms should be observed until the next access. With some EEPROMs, Read/Busy information is available on pin 1. If such a device is employed, then their status may be interrogated by a program reading the ERDY1/ERDY2 (EEPROM-Ready) lines.

Decoder IC6 supplies the chip-select signals for the clock device, IC5, the watchdog in the MAX690 (IC3) and the extension bus.

Table 1. Minimum configuration of C166 board

R7	resistor, 1 kΩ	Pull-up for $\overline{S0}$
R9	resistor, 10 kΩ	Pull-up for $\overline{WR}$
R23	resistor, 4.7 kΩ	Pull-up for $\overline{RSTINT}$
R24	resistor, 10 kΩ	Pull-up for $\overline{NMI}$
C1,C2	capacitor, 10 µF, tantalum	Decoupling
C9,C10,C16,C17	capacitor, 100 nF	Decoupling
C22-C25	capacitor (tantalum), 10 µF	SIO 2
T1	transistor, BS170	For S0 signal (Chip Select 0 for RAM)
IC1	MAX232	SIO 2
IC4	GAL 16V8, 15-12 ns	Programmed
IC8	SAB 80C166M	Microcontroller
IC9/IC10	74HC573	Address latch
IC11/IC12	62256	Static RAM
IC13/IC14	27C256	EPROM (user program or system monitor)
Q1	40 MHz	Oscillator or quartz crystal w. C18/C19
JP1,JP2,JP5		

Table 2. Modified contents of configuration file

```
;****************************************************************
;*    configuration file for EVA165/166/167 monitor MON16X      *
;*                      (c)ertec GmbH                           *
;****************************************************************
bios=0                   ; use direct port i/o
;baud=9600               ; baudrate 9600
baud=38400               ; baudrate 38400
com=2                    ; use COM2
;boot=boot.166           ; loader for monitor MON166
;monitor=mon.166         ; monitor MON166
;****************************************************************
```

SOLDER A CENTIPEDE

Most of you will agree that soldering the 100-pin microcontroller on to the board is a daunting task. Alas, there is no alternative because the 80C166 is only supplied in a 100-pin MRFP case.

Apart from fine-gauge solder (max.

1 mm dia.) for fitting the ‘regular’ parts you will need suitable solder cream and solder flux. The solder iron should have a fine tip, and be temperature controlled at about 320 °C. The microcontroller is the first part to be soldered on to the board. Before you start soldering, make sure that all 100 pins are properly aligned to the pads on the PCB. If necessary, carefully adjust their position. Next, mix five or six parts of solder flux with one part of solder cream, and dilute this with a drop of methylated spirit until the mixture starts to become liquid. The mixing is conveniently done in a beer bottle cap, using a toothpick as a stirring device. The mixture is then carefully applied to the edges of the solder pads, or to the ends of the microcontroller pins. This is best done using a very small brush. In general, the less mixture you apply, the better.

Next, the controller may be positioned on the pads again, and fixed in position by soldering two diagonally located pins. Now once again check the correct alignment of each and every pin over its own solder pad. If no obvious problems are detected, you can start soldering. Using one finger to push the controller firmly on to the board and at the same time check its temperature, carefully and slowly pull the tip of the solder iron past the pin extremes (i.e., not across the pins). Provided the pad and the pins are uniformly heated, the solder cream will also contract and solidify in a uniform manner through the solder flux. When you feel that the chip gets too hot, take a short break, and continue where you left off.

Use a magnifying glass to inspect all solder joints, and correct any one that needs to be brought in line. If an unwanted joint has been made, press desoldering braid and some flux on to the link, heat the joint and remove the excess solder by a gradual pulling motion. Flux and solder cream residue may be removed with a cotton swab dipped in methylated spirit.

With the 80C166 safely ensconced on the board, construction may proceed, starting the lower-profile parts (resistors), and, finally, the taller parts like the electrolytic capacitors. The parts list, incidentally, indicates a minimum configuration (M) for the board, see also **Table 1**.

As a replacement for IC3 (MAX690), connect a 10-μF tantalum electrolytic capacitor to IC3 pin 7 (+ wire) and pin 3 (– wire). This capacitor will act as

1

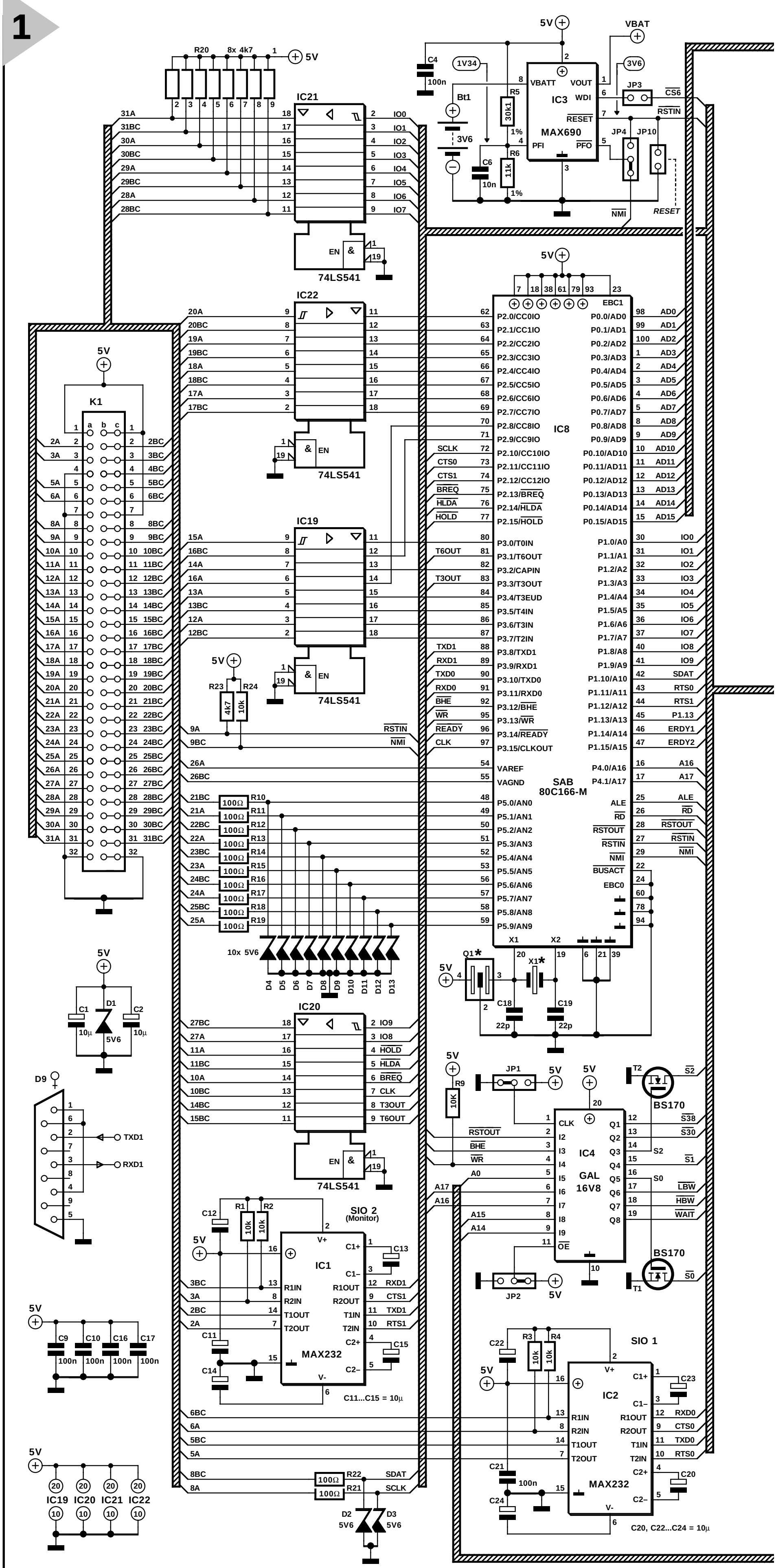
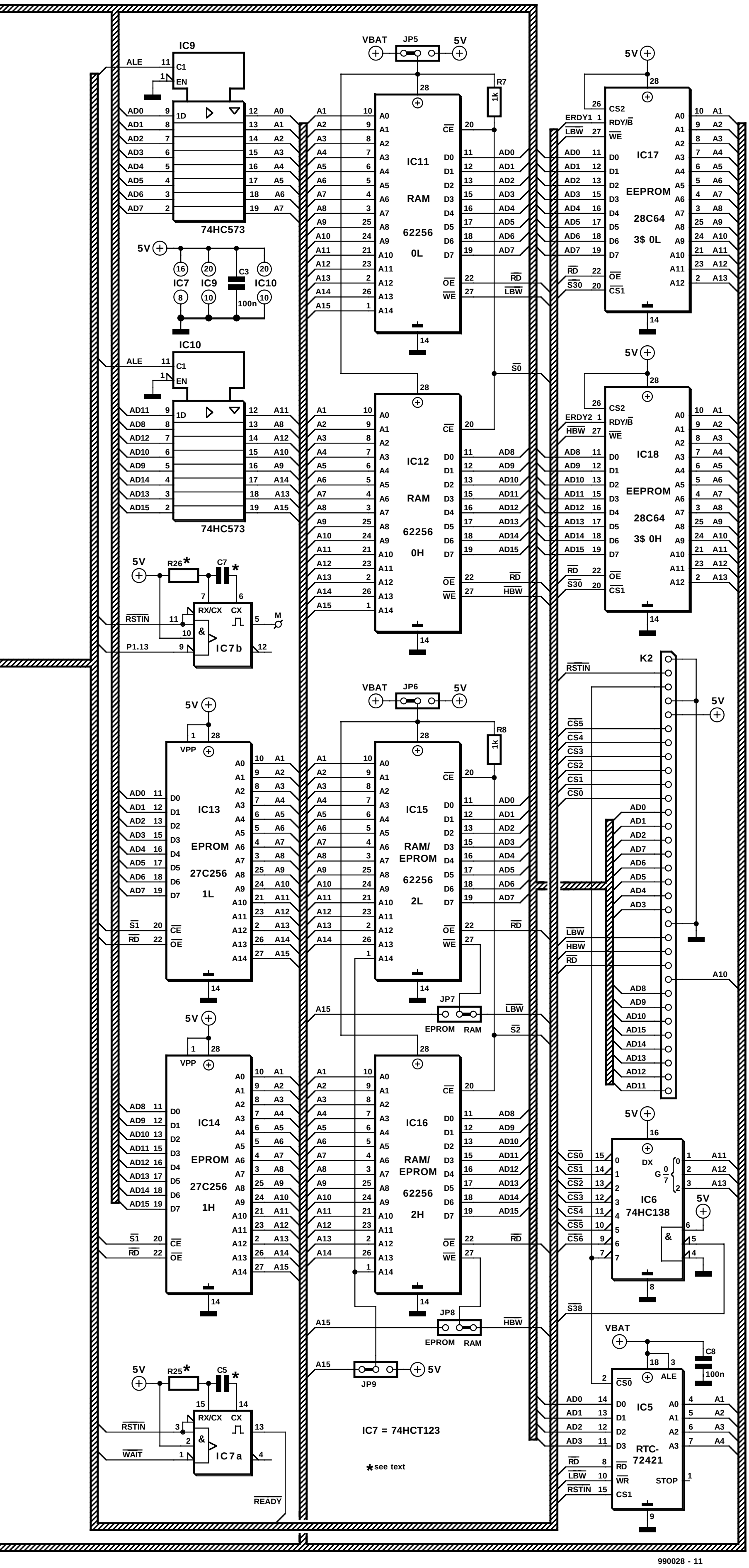


Figure 1. The quite extensive circuit diagram of the 80C166 16-bit microcontroller evaluation system.



the power-on reset. When the MAX690 is employed, jumper JP3 should be pulled. If not, the MAX690 will faithfully produce a reset signal at 1-second intervals.

To power the RAM chips, install JP5 in the 5-V position, and connect GAL pins 1 and 11 to ground using JP1 and JP2.

When you are satisfied with your construction efforts, it is time to give the board a last and thorough visual inspection. If no faults are discovered, the ICs may be inserted into their sockets.

POWERING-UP

Connect the 80C166 board to a suitable 5-volt dc power supply whose current limit is set to 0.3 A. Switch on, and check that the current consumption settles at about 0.2 A.

ENTER YOUR PC

To enable the 80C166 board to communicate with a PC, it has to be fitted with two EPROMs (IC13 for the low byte and IC14 for the high byte) containing the monitor program. On the PC, the program `MON16X.EXE` or `MON16X2.EXE` has to be installed with the files `MON16X.OVL`, `MON16X.CFG` or `MON16X.XSR`. Connect the serial cable between SIO2 and a free communication port (RS232) on your PC, and then launch the program `MON16X.EXE`. Two parameters should be included:

```
[Drive:\path\]MON16X
-c [Drive:\path\]MON16X.CFG
-s [Drive:\path\]MON16X.SCR
```

where
`-c` indicates the drive/path/name of te configuration file;
`-s` indicates the drive/pathname of the script file.

To keep things as simple as possible, the program should be launched with the aid of batch file.

The contents of the original `MON16X.CFG` configuration file has to be modified as shown in **Table 1**. The lines in **bold** print are new, while those that are no longer required have a semicolon (;) in front of them.

The script file, `MON16X.SCR`, could have the following contents:

```
key 1 “Help” A
key 4 “S” A
key 5 “PROC” A
key 9 “EXIT” A
stat
```

The PC program first transmits a dummy byte to enable the monitor program on the board to detect the baud rate. The board returns 0AAh to

2

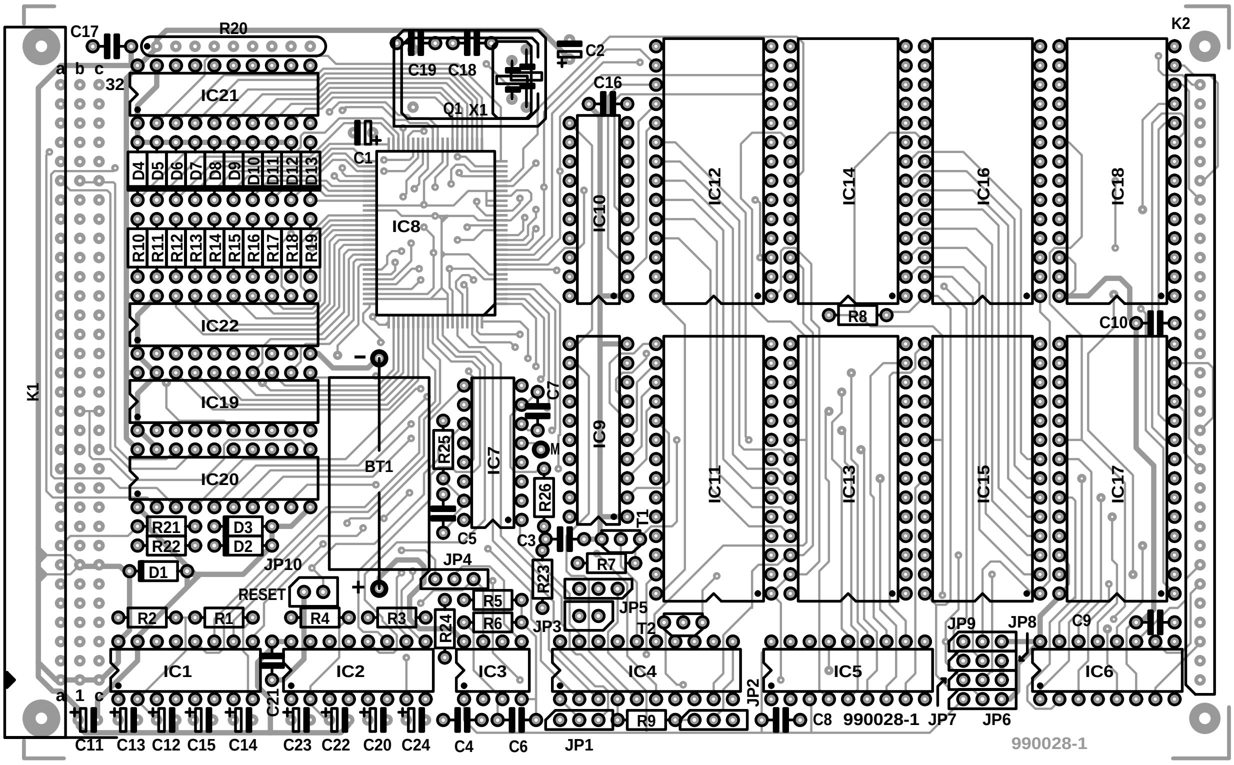


Figure 2. Densely packed: the double-sided, through-plated, Eurocard-sized PCB for the evaluation system.

indicate that it is ready. The connection with the PC is ready for use after the version texts and the final prompt **MON166>** have appeared.

Once the monitor program is ‘on line’, programs may be uploaded to the 80C166 board. This is done with the command ‘L’ (for Load), followed by the filename, a 80C166 program in hexadecimal format, or as an ‘absolute ‘file:

```
L
“[Drive:\path\]filename.hex”
or
L “[Drive:\path\]file[.abs]”
```

Using the command `G` (Go) and the relevant start address, the previously loaded program is executed.

The command `H` is available to enable all monitor commands to be listed. The monitor program also allows further hardware tests and memory checks to be run.

If, after some time, you see one of the following error reports on the PC display: *serial send error*, *communication error…*, *timeout*, *fatal error … abort*, or similar, you have to get ready for a faultfinding session.

HARDWARE DEBUGGING

To begin with, inspect all solder joints, particularly those on the microcontroller. Also check that all parts have been fitted in the right PCB positions. A correctly operating serial link is also essential.

Here are a few hardware-related problems you may come across:

COMPONENTS LIST

Resistors:
R1-R4 = 10kΩ (SIO1)
R5 = 30kΩ 1% (I/O)
R6 = 11kΩ 1% (I/O)
R7 = 1kΩ (M)
R8 = 1kΩ (RAM/ROM)
R9,R24 = 10kΩ (M)
R10...R19,R21,R22 = 100Ω (I/O)
R23 = 4kΩ7 (M)
R25,R26 = see text (RDY)

Capacitors:
C1-C4,C8,C9,C10,C16,C17,C21 = 100nF ceramic, raster 5mm (M)
C5,C7 = see text
C6 = 10nF raster 5mm (I/O)
C11-C15 = 10µF 16V radial (M)
C18,C19 = 22pF (only with 100MHz-crystal)
C20,C22,C23,C24 = 10µF 16V radial (SIO1)

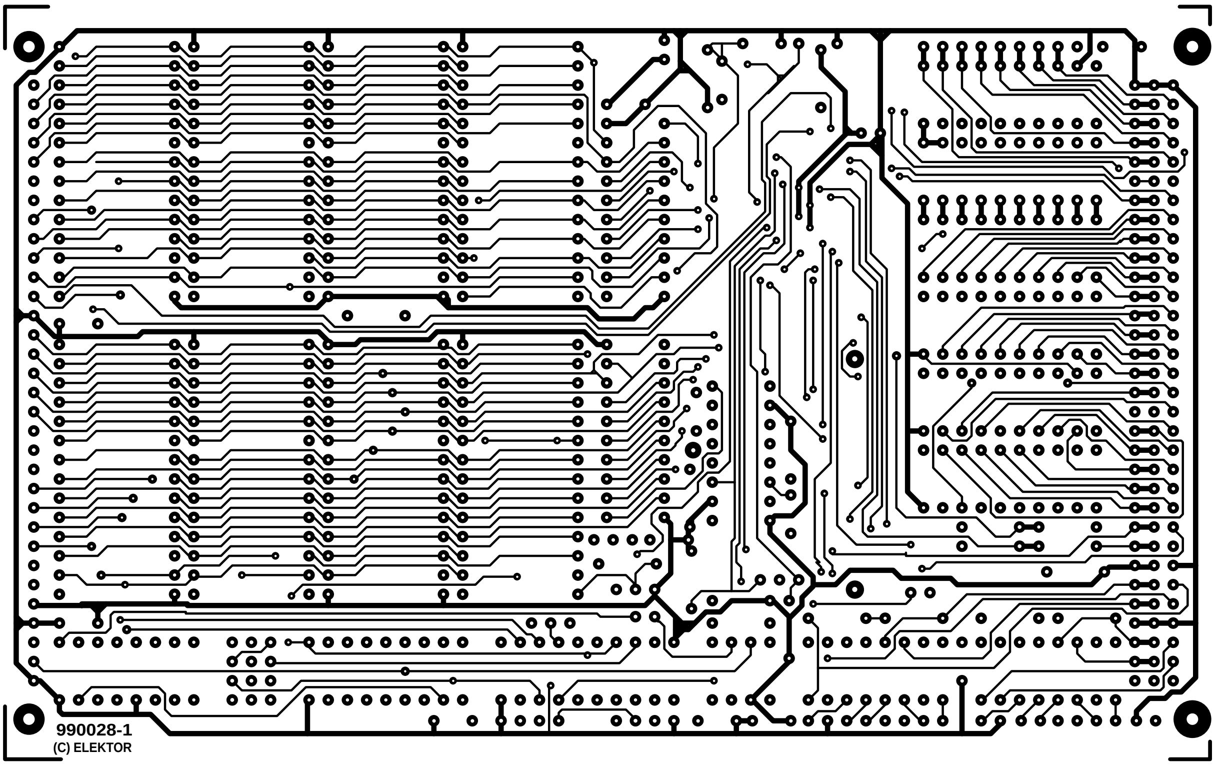
Semiconductors:
D1 = ZD 5V6/1W3 (M)
D2-D13 = zener diode 5.6V 1.3W (I/O)
T1 = BS170 (M)
T2 = BS170 (RAM/ROM)
IC1 = MAX 232 (M)
IC2 = MAX 232 (SIO1)
IC3 = MAX 690 (I/O)
IC4 = GAL 16V8-15 (programmed, 996508-1) (M)
IC5 = RTC 72421 (RTC)
IC6 = 74HCT138 (BATT, RTC)
IC7 = 74HCT123 (RDY)
IC8 = SAB 80C166 M (M)
IC9,IC10 = 74HCT573 (M)
IC11,IC12 = RAM 62256-100 (M)
IC13= EPROM 27C256-120 (Low Byte, programmed, 996509-1) (M)
IC14 = EPROM 27C256-120 (High Byte, programmed, 996509-2) (M)
IC15,IC16 = RAM 62256-100 or EPROM 27C256-120 (RAM/ROM)
IC17,IC18 = EEPROM 28C64-250 (EEPROM)
IC19-IC22 = 74LS541/540 * (I/O)

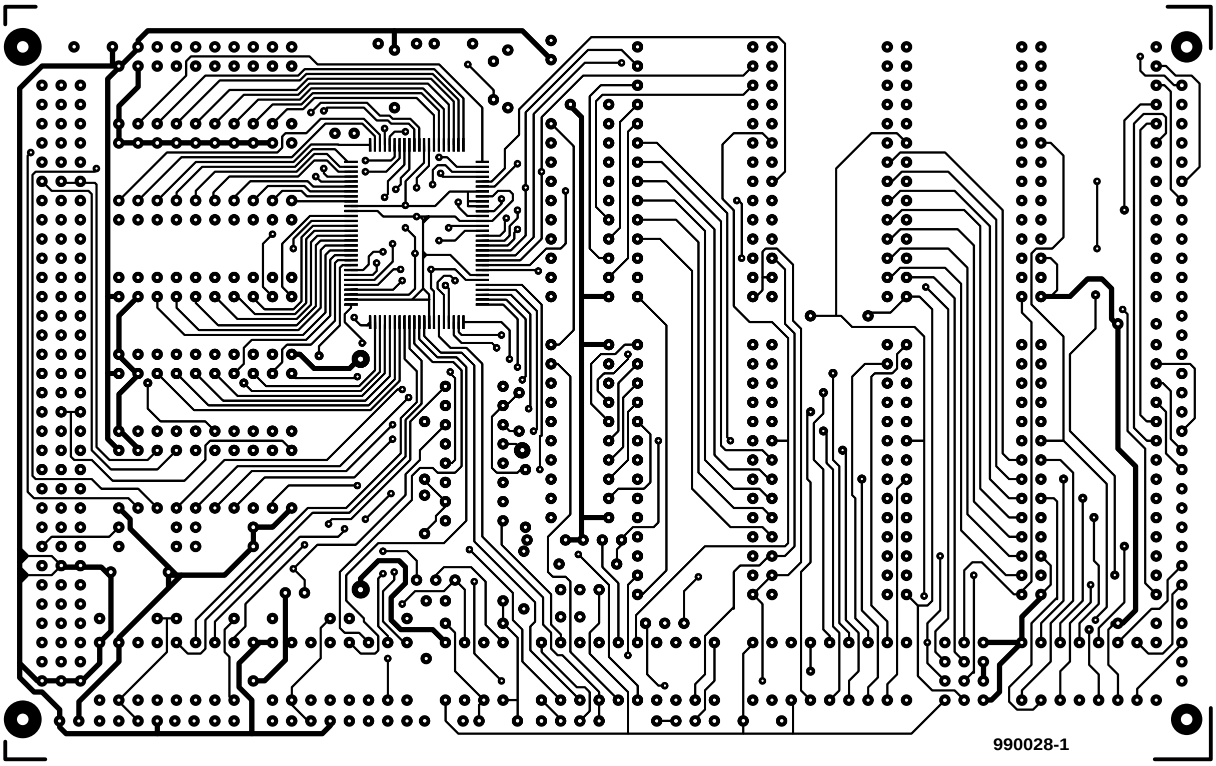
Miscellaneous:
Q1 = oscillator module 40MHz (M)
IC sockets, 8/16/20/28-pin (M)
IC socket, 18-pin (RTC)
64-way pinheader or DIN-connector 41612 (M)

Items available through the Publishers’ Readers Services:

996509-1, -2:	programmed EPROMs
996508-1:	programmed GAL
996009-1:	disk, C166 monitor (compiled and source code), and GAL listing
996009-2	disk, high-level debugger from Hitex
990028-1:	Printed circuit board

(M)	Minimum configuration
(I/O)	Input/Outputs
(BATT)	MAX690, battery
(SIO1)	SIO1 extension
(RTC)	Real-time clock extension
(RAM/ROM)	RAM or EPROM extension
(EEPROM)	EEPROM extension
(RDY)	READY extension





- Wrong serial port, or wrong entry in the PC configuration file `MON16X.CFG`.

- RxD and TxD lines to PC swapped.

- No interconnections between handshaking lines CTS and RTS, DTR, DSR and CD, on the PC interface. These links are needed to inform the PC that a DTE (data terminal equipment, i.e., the 80C166 board) is connected to the serial port.

- Missing (monitor) program in EPROMs, or 'low' and 'high' EPROMs swapped.

- GAL (IC4) not fitted, not programmed or incorrectly programmed.

- JP1 and/or JP2 (pins 1 and 11 of the GAL) incorrectly installed.

- Missing SIO (IC1) or electrolytic capacitors required for correct functioning.

- Quartz crystal or oscillator module does not oscillate (measure at Q1 pin 3).

- Reset input remains low. When the MAX690 (IC3) is used without the battery, reset may remain low (measure at RESET pin 7 IC3 or K1 pin 9a).

- Output $\overline{\text{RSTOUT}}$ remains low. Dur-

Table 4. Contents of file 80C166.EQN

```
; GAL for 80C166 Controller Board
chip 80C166 GAL16V8
; Pin  1     2      3      4     5      6     7      8     9      10
       j1    RST    BHE    WR    A0     A17   A16    A15   A14    GND
       j2    S38    S30    S2    S1     S0    LBW    HBW   WAIT   VCC
; Pin  11    12     13     14    15     16    17     18    19     20
equations
S0           = /A17 * /A16 *  RST               ; RAM               0$0000h-0$FFFFh
             + /A17 * /A16 * /RST *  j2         ; (with /RST)       0$0000h-0$FFFFh
/S1          = /A17 * A16                       ; EPROM             1$0000h-1$FFFFh
             + /A17 * /A16 * /RST * /j2 * /j1   ; (with /RST)       0$0000h-0$FFFFh
S2           =  A17 * /A16                      ; RAM/EPROM         2$0000h-2$FFFFh
             + /A17 * /A16 * /RST * /j2 * j1    ; (with /RST)       0$0000h-0$FFFFh
/S30         =  A17 *  A16 * /A15               ; EEPROM            3$0000h-3$8000h
/S38         =  A17 *  A16 *  A15 * /A14        ; Decoder           3$8000h-3$BFFFh
/LBW         = /WR  * /A0                       ; Low Byte Write
/HBW         = /WR  * /BHE                      ; High Byte Write
; Optional
/WAIT        = A17 * A16 * /A15                 ; EEPROM            3$0000h-3$8000h
             + A17 * A16 *  A15 * /A14          ; Decoder           3$8000h-3$BFFFh
```

ing a reset, $\overline{\text{RSTOUT}}$ should drop low, and return to high as soon as the reset condition is ended. If that does not happen, the (monitor) program is simply not executed, and the controller will never reach the EINIT (End Init) instruction. EINIT returns the $\overline{\text{RSTOUT}}$ output to high (measure at IC4 pin 2).

- Address Latch Enable (ALE) signal not available. No matter if the controller contains a fully debugged award-winning program, or just fetches nonsense, memory access will occur every 150 to 850 ns, with the ALE signal being pulled high for about 20 ns (measure at IC9/IC10 pin 11).

- Missing or wrong address bits. Address bits 0-15 are latched on every ALE pulse. The latch components IC9 and IC10 have to be fast enough to recognize the short ALE signal, and faithfully copy the address bits which are available for about 40 ns (measure at IC9/IC10 pins 12-19).

- Missing chip-select signal on EPROMs (measure at IC13/IC14 pin 20). The chip select signals for the memory devices are generated by the system GAL (IC4). The bit combinations shown in **Table 3** are required to for combining into the 'chip select-1' signal for the EPROMs (pin 15 on IC4). The bits should be measured on GAL IC4. During a reset, all lines are at high impedance (i.e., at 'input').

Table 3. CS1 on GAL

After reset	After EINIT
$\overline{\text{RSTOUT}}$ (Pin 2) = 0	$\overline{\text{RSTOUT}}$ (Pin 2) = 1
A17 (Pin 6) = 0	A17 (Pin 6) = 0
A16 (Pin 7) = 0	A16 (Pin 7) = 1
A15 (Pin 8) = 0	A15 (Pin 8) = 0
A14 (Pin 9) = 0	A14 (Pin 9) = 0
$\overline{\text{S1}}$ (Pin 15) = 0	$\overline{\text{S1}}$ (Pin 15) = 0

- No serial connection. Each time MON16X is started, you should be able to measure serial transmission signals on the RxD line (IC1 pin 13/12). The monitor program replies via the TxD line (IC1 pins 11/14). Measure at IC1 pins 13, 12, 11 and 14).

- Simply doesn't work…? Your construction may contain other errors or

Pin functions, extension

Pin	Name	Function
1	GND	Supply ground
2	$\overline{\text{RSTIN}}$	Restart input (Reset)
3		
4	GND	Supply ground
5	+5	+5 V supply
6	$\overline{\text{CS5}}$	Chip select 5: 3A800-3AFFFh
7	$\overline{\text{CS4}}$	Chip select 4: 3A000-3A7FFh
8	$\overline{\text{CS3}}$	Chip select 3: 39800-39FFFh
9	$\overline{\text{CS2}}$	Chip select 2: 39000-397FFh
10	$\overline{\text{CS1}}$	Chip select 1: 38800-38FFFh
11	$\overline{\text{CS0}}$	Chip select 0: 38000-387FFh
12	AD0 / P0.0	Address/data bit 0
13	AD1 / P0.1	Address/data bit 1
14	AD2 / P0.2	Address/data bit 2
15	AD7 / P0.7	Address/data bit 7
16	AD6 / P0.6	Address/data bit 6
17	AD5 / P0.5	Address/data bit 5
18	AD4 / P0.4	Address/data bit 4
19	AD3 / P0.3	Address/data bit 3
20	GND	Supply ground
21	$\overline{\text{LBW}}$	Low Byte Write
22	$\overline{\text{HBW}}$	High Byte Write
23	$\overline{\text{RD}}$	Read
24	A10	Address bit 10
25	AD8 / P0.8	Address/data bit 8
26	AD9 / P0.9	Address/data bit 9
27	AD10 / P0.10	Address/data bit 10
28	AD15 / P0.15	Address/data bit 15
29	AD14 / P0.14	Address/data bit 14
30	AD13 / P0.13	Address/data bit 13
31	AD12 / P0.12	Address/data bit 12
32	AD11 / P0.11	Address/data bit 11

a defective part you have not noticed so far. Further investigation is required.

FREEWARE AND MORE

The hardware of the 80C166 evaluation system should be complemented by a range of software products. The aim of the discussion below is to prevent confusion arising about this particular subject.

Basically, a distinction has to be made between software on the board and software running on the PC.

On the board, the monitor program is permanently stored in two EPROMs. This monitor program was specially developed by the author for the 80C166 board. It is faster, better and less bug-ridden than comparable, non-adapted products. Its only disadvantage is that the EPROM pair reduces the memory range by 64 kBytes. Unfortunately, the 80C166 does not support bootstrap loading, this is only possible on the 80C165 and 80C167. The monitor program is supplied in two ready-programmed EPROMs (Publishers' order codes **996509-1** and **996509-2**). Alternatively, burn your own EPROMs using the hex file found on the project diskette (Publishers' order code **996091-1**).

The GAL chip may also be obtained ready-programmed (order code **996581-1**). However, if you have a programmer for the GAL16V8, you may want to use the GAL equations file listed in **Table 4**.

The `MON16X` program for use on the PC was not developed by the author. It is supplied as shareware or test software on CD-ROM or through the Internet, by a number of manufacturers of microprocessor-related products, including our advertiser Hitex. See, for example, these web sites:

www.hitex.com
www.keil.com
www.tasking.com
www.ertec.com

With nearly all suppliers, C or C+ is the 'dominant' programming language.

The author uses the Keil assembler/linker/locator (which is not free of charge), in combination with collected software from ertec GmbH, packed in the file **EVAAL.ZIP** (found in the 'download' area on ertec's web site). Actually, this software is intended for ertec's type EVA166 development board, which is compatible with the evaluation system described in this article series. The zipped file contains, among others, these programs which are of special interest to us:

MON16X2.EXE	Windows 95 program
MON16X.EXE	MS-DOS program
MON16X.OVL	overlay for MON16X
MON16X.SCR	script file for things to be automatically done by MON16X or MON16X2
MON16X.CFG	configuration file for launching MON16X or MON16X2
ASS16X.EXE	C166 assembler for DOS
ASS16X2.DLL	C166 assembler for Windows 95

The bootstrap loader has to be disabled in the configuration file by putting semicolons in front of the relevant command lines. The EPROM-resident version of the monitor as developed by the author achieves a baud rate of 38,400 bits/s.

Finally, a vast number of application examples, FAQs and links to other information on the 'C166 micro may be found at
www.hitex.demon.co.uk/c166/miscdocs.html

(990028-2)

Pin functions, controller board

Function		Name	Pin		Pin		Name	Function	
+5V	Supply	**+5V**	1bc	1	2	1a	**+5V**	+5V	Supply
Output	SIO 2 TxD Mon	P3.8 / TXD1	2bc	3	4	2a	P1.12	Output	SIO 2 RTS
Input	SIO 2 RxD Mon	P3.9 / RXD1	3bc	5	6	3a	P2.12/CC12IO	Input	SIO 2 CTS
GND	SIO 2	**GND**	4bc	7	8	4a	GND	GND	SIO 1
Output	SIO 1 TxD Prog	P3.10 / TXD0	5bc	9	10	5a	P1.11	Output	SIO 1 RTS
Input	SIO 1 RxD Prog	P3.11 / RXD0	6bc	11	12	6a	P2.11/CC11IO	Input	SIO 1 CTS
+5V	Serial	+5V	7bc	13	14	7a	GND	GND	Serial
*Input/Output Serial Data **		P1.10	8bc	15	16	8a	P2.10 /CC10IO	*Input/Output Serial Clock**	
Input	Non Maskable Int	NMI	9bc	17	18	9a	**RSTIN**	Input	Reset
Output	(20 MHz Clock)	P3.15 / CLK	10bc	19	20	10a	P2.13 / BREQ	Output	(Bus Request)
Output	(hold acknowledge)	P2.14 / HLDA	11bc	21	22	11a	P2.15 / HOLD	Output	(Hold)
Input	(Timer 2 Input)	P3.7 / T2IN	12bc	23	24	12a	P3.6 / T3IN	Input	(Timer 3 Input)
Input	(Timer 4 Input)	P3.5 / T4IN	13bc	25	26	13a	P3.4 / T3EUD	Input	(Timer 3 Up/Dn)
Output	(Timer 3 Output)	P3.3 / T3OUT	14bc	27	28	14a	P3.2 / CAPIN	Input	(Capture Input)
Output	(Timer 6 Output)	P3.1 / T6OUT	15bc	29	30	15a	P3.0 / T0IN	Input	(Timer 0 Input)
Input	(Capt/Comp) 10	P2.9 / CC9IO	16bc	31	32	16a	P2.8 / CC8IO	Input	(Capt/Comp) 9
Input	(Capt/Comp) 8	P2.7 / CC7IO	17bc	33	34	17a	P2.6 / CC6IO	Input	(Capt/Comp) 7
Input	(Capt/Comp) 6	P2.5 / CC5IO	18bc	35	36	18a	P2.4 / CC4IO	Input	(Capt/Comp) 5
Input	(Capt/Comp) 4	P2.3 / CC3IO	19bc	37	38	19a	P2.2 / CC2IO	Input	(Capt/Comp) 3
Input	(Capt/Comp) 2	P2.1 / CC1IO	20bc	39	40	20a	P2.0 / CC0IO	Input	(Capt/Comp) 1
Input.	*(Analogue) 1 **	P5.0 / AN0	21bc	41	42	21a	P5.1 / AN1	Input	*(Analogue) 2 **
Input.	*(Analogue) 3 **	P5.2 / AN2	22bc	43	44	22a	P5.3 / AN3	Input	*(Analogue) 4 **
Input	*(Analogue) 5 **	P5.4 / AN4	23bc	45	46	23a	P5.5 / AN5	Input	*(Analogue) 6 **
Input.	*(Analogue) 7 **	P5.6 / AN6	24bc	47	48	24a	P5.7 / AN7	Input	*(Analogue) 8 **
Input	*(Analogue) 9 **	P5.8 / AN8	25bc	49	50	25a	P5.9 / AN9	Input	*(Analogue) 10 **
Analogue Ground (GND)		VAGND	26bc	51	52	26a	VAREF	Analogue Reference (+5V)	
Output	10	P1.9	27bc	53	54	27a	P1.8	Output	9
Output	8	P1.7	28bc	55	56	28a	P1.6	Output	7
Output	6	P1.5	29bc	57	58	29a	P1.4	Output	5
Output	4	P1.3	30bc	59	60	30a	P1.2	Output	3
Output	2	P1.1	31bc	61	62	31a	P1.0	Output	1
GND	Supply	**GND**	32bc	63	64	32a	GND	**GND**	Supply